

1 **A USER INTERFACE MECHANISM FOR MANIPULATING CONTEXT**
2 **IN COMPUTER MANAGEMENT APPLICATIONS**
3
4

5 The instant application is a continuation-in-part application of copending
6 patent application serial number 09/087,066 filed on May 29, 1998.
7

8 **CROSS REFERENCES TO RELATED APPLICATIONS**

9 The subject matter of the present application is related to copending
10 United States application, Serial No. 09/086,635, titled "A Method and System
11 to Provide Functionality Access Based on User Approach to Network and
12 System Management Tasks", Docket No. 10980425-1, filed May 29, 1998
13 contemporaneously with the copending parent application (serial number
14 09/087,066) of the instant CIP application and issued on January 16, 2001 as
15 U.S. Patent No. 6,175,363, and copending United States application, Serial
16 No. 09/087,583, titled "A User Interface Mechanism for Maintaining Quick
17 Access to Important Information in a Windows-Based Computer Environment",
18 Docket No. 10980426-1, filed May 29, 1998 contemporaneously with the
19 copending parent application (serial number 09/087,066) of the instant CIP
20 application and issued on March 28, 2000 as U.S. Patent No. 6,043,816, both
21 of which are assigned to the assignee hereof and are herein incorporated by
22 reference.
23

24 **FIELD OF THE INVENTION**

25 The present invention relates generally to windows-based computer

1 applications, and more particularly to a user interface mechanism for
2 manipulating context in windows-based computer applications, such as
3 network and systems applications.
4

5 BACKGROUND OF THE INVENTION

6 Many computer environments employ windows having graphical user
7 interface (GUI) capabilities that aid the user in interacting with various
8 applications in a straight-forward and intuitive manner. Whether the user is
9 operating a stand-alone computer, a networked computer, or other equipment
10 in the computer environment, graphical user interfaces presented in windows
11 have become the environment of choice for many users rather than command
12 line interfaces. The windows-based computer environment enables the user
13 to access any number of computer applications simply by opening an
14 appropriate window for the desired application.
15

16 Typically a user operating in a windows-based computer environment
17 must open a new window each time that a new application is to be used.
18 Many applications do not have all of the functionality needed to accomplish all
19 aspects of a user's job (e.g. editing graphics). Assume that a user has two
20 graphical editing applications, one application provides editing capabilities for
21 vector graphics and the other provides editing of pixel-based graphics. If the
22 user has been editing vector graphics with one application and now wants to
23 edit pixel-based graphics, the user must first open a second graphics editing

1 application in a new window. It is not currently possible to access the pixel-
2 based graphics editing application from the open application for editing vector
3 graphics since a separate window must be opened for each application. There
4 is therefore a requirement of a one-to-one correspondence between a
5 computer application and a window that supports the application. For anyone
6 using a sizable number of windows-based applications, this means that the
7 user must open a great number of windows.

8
9 A disadvantage of this requirement is that any open window must be
10 managed by a window manager of the graphical user interface (GUI) that
11 produces and drives windows on a display. A window manager is a software
12 program or module that defines the parameters of windows and generates the
13 windows. The more windows that are open in a computer environment, the
14 more that overhead associated with window manager programs will slow down
15 the operation of the environment. The excess of windows also creates
16 cognitive overhead for the user in that they have to deal with minimizing,
17 restoring, and moving windows to be able to interact with the separate
18 applications.

19
20 Another common characteristic of graphical user interfaces, particularly
21 for those addressing network systems and service management, is that they
22 primarily provide a single view of objects and a single set of capabilities for
23 manipulating the objects they present. Typically there is no means to see

1 objects from different perspectives. For example, if a user is accessing an
2 application to look at the configuration of a computer, they cannot easily see
3 that same computer from the perspective of where it sits in a network and how
4 it communicates with other computers. To do this users typically need to bring
5 up separate applications and take actions to locate the desired object in both.
6 And if the user can see the objects from different perspectives, the different
7 perspectives and related functionality are typically intermixed causing an
8 overload of functionality.

9
10 A somewhat less common aspect of graphical user interfaces is the use
11 of information about the current context to aid the user in performing tasks.
12 For instance, information on where the pointer is located when the mouse is
13 clicked is used to filter pop-up menus. Information on the pointer position or
14 specific location in an application is used to provide context sensitive help.
15 However, there has been no attempt to use context at a higher level to provide
16 users with different information sets or different perspectives on a given
17 information set. User task performance would greatly benefit from an explicit
18 mechanism by which users can designate the perspective they want to take in
19 interacting with a set of applications. Applications could then use this
20 contextual information to determine the type of data and functionality to be
21 presented, the manner in which the data will be presented (e.g., different
22 layouts based on the types of object relationships that are associated with the
23 context), and the default parameters to be used (e.g., systems may be polled

1 every 30 minutes in a system context to make sure they are up and running;
2 however, in an application context they may be polled every 10 minutes to
3 determine if the application is still running) .
4

5 There is therefore an unmet need in the art to be able to minimize the
6 number of windows that are required for related applications or application
7 functionality. Additionally, there is an unmet need in the art to allow users to
8 designate a context and have applications share relevant contextual
9 information in such a way that will allow for the presentation of data and
10 functionality based on the perspective associated with that context.
11

SUMMARY OF THE INVENTION

According to the present invention, a method provides contextual information to a user of a windows-based computer environment. This method comprises opening a window having a user interface and manipulating a context control feature of the user interface to determine a first context of the window corresponding to at least a first perspective of one or more perspectives for viewing and manipulating one or more objects, with each object of the one or more objects having one or more object capabilities within the window. The first perspective presents a filtered view of at least one of the one or more object capabilities of the one or more objects corresponding to the first perspective.

Also in accordance with the present invention, a user interface of a window provides contextual information to a user of a windows-based computer environment. The user interface comprises a context control feature and a content pane. The context control feature has one or more available contexts operable to present one or more corresponding perspectives within the window. Manipulating the context control feature of the user interface determines a context of the window from the one or more available contexts corresponding to at least a first perspective of the one or more perspectives for viewing and manipulating a plurality of objects. Each object of the plurality of objects has one or more object capabilities that are illustrated in the content

1 pane of the user interface of the window. The first perspective presents a
2 filtered view of at least one of the one or more object capabilities of the one or
3 more objects corresponding to the first perspective.
4
5

100200122-17

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the claims. The invention itself, however, as well as the preferred mode of use, and further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawing(s), wherein:

Figures 1a and 1b illustrate the use of a context list box in an application window that presents views of objects and provides functionality for acting on those objects, according to one embodiment of the present invention;

Figures 2a, 2b, 2c and 2d illustrate the use of a context list box in an application window that allows users to access different types of objects or applications, according to one embodiment of the present invention; and

Figures 3a and 3b illustrate the use of a context list box in an application window that provides lists of tasks and allows users to access the user interface for accomplishing those tasks, according to one embodiment of the present invention.

Figures 4a and 4b illustrate the impact of contextual perspectives on the functionality available for manipulating the same object in two or more

1 different contexts each having a different perspective, according to one
2 embodiment of the present invention. **Figure 4c** illustrates a mechanism by
3 which manipulations associated with other perspectives can be made available
4 to the user while keeping the manipulations relatively hidden, according to one
5 embodiment of the present invention.

6

100200122-1 9

DESCRIPTION OF THE INVENTION

Contextual information is often used to filter information in the user interface of a windows-based computer environment. It is thus useful as a filtering tool to provide information to the user that is consistent with the objects or applications the user is currently using or the perspective from which the user is viewing the objects or applications. The present invention employs the user interface of a single open window, such as an application window, to empower the user to assert direct control over the filtering of information and the perspective from which information is presented to the user in the single window. Through the manipulation of a context control feature of the user interface, the user can choose the desired context for the window. The context is defined such that the objects that appear in the window, their relationships, and capabilities for manipulating the objects are tied to the context. For instance, using the present invention, the user may see the network connectivity of a computer system or the IT services (e.g., email, file sharing) in which the system participates all inside a single window by correctly manipulating the context control feature of the window. The context of the window may be readily changed as desired, again through appropriate manipulation of the context control feature.

The context control feature, which may be a context list box, a context tab, or other control feature of the user interface, contains a list of the contexts available within the window. User manipulation of context through the context

1 control feature of the window is accompanied by a corresponding change in
2 menu items, toolbar buttons, and views of presented objects according to the
3 context chosen by the user. Relevant information from the initial context, such
4 as currently selected objects, is maintained in the new context of the window
5 and can be used to aid users in quickly accessing task relevant information or
6 functionality. This feature makes the user more efficient in the task at hand.

7
8 Manipulation of the context control feature of the GUI of a single window
9 can enable the user to look at a given set of managed objects in a different
10 perspective or context. Alternatively, changing context can mean the user will
11 have access to a different set of objects. Thus the user can look at a different
12 set of objects from the same perspective as an earlier set of objects.

13
14 First, consider the use of the present invention as an explicit mechanism
15 for filtering information and changing the perspective from which information is
16 presented to the user via the user interface. A change in context via the
17 context control mechanism is typically associated with a change in one or
18 more of the objects presented and the type of capabilities available or visible
19 to the user for manipulating those objects.

20
21 Consider, for example, an application window in which the user can
22 change between the context for the IP Network Management and the context
23 for Service Management using the context control mechanism of the context

list box. This use of the context list box is illustrated in Figures 1a and 1b. Referring to **Figure 1a**, the context list box 12 of the user interface 10 has been set to "IP Internet" and the location cursor is on "Segment 1" in scoping pane 14 so that the context of user interface 10 is IP Internet. The content pane 16 illustrates the IP Internet perspective of objects of Segment 1 of the IP Internet: CltGeorg, nsmdedd-green, et2m41, djm213t, djm23t, and ljm21; note that objects CltGeorg, nsmdedd-green, djm213t, djm23t, and ljm21 are all connected to one another through the network hub et2m41 as indicated by the graphical map configuration shown in content pane 16. In **Figure 1b**, the same application window with a different context is presented to the user. Context list box 12 has been set to "IT Services" and scoping pane 14 is set to show different types of services so that user interface 20 has a Service Management context. Content pane 16 illustrates the objects involved in the OpenMail service. Note that the selected object "CltGeorg" from user interface 10, has been carried over to show this same object from the new perspective of Service Management. This system is now connected to the mail server "Mail-SV1" that provides email to the system. Notice that available menus 17 and tool bar buttons 18 have been changed to match the new context. By simply manipulating the context chosen in context list box 12, the user explicitly filters information and changes the perspective from which information concerning managed objects is presented to the user.

Now consider an example in which context is used to view different

1 objects or applications. This use of the context control is illustrated in Figures
2 2a, 2b, and 2c. In **Figure 2a**, the context chosen in the application window of
3 user interface 30 is "Users" as shown in context list box 12 and scoping pane
4 14. In this example, "Users" is the only visible choice in the scoping pane but
5 this item could be expanded by the user to display additional choices. The
6 objects illustrated in content pane 16 include the following user groups: admin,
7 operator, user, MS-SQL, and MS-Exchange. These are the groups of users
8 that are allowed to do backup and recovery of files within the application.

9
10 Typically, using the context control feature, a user will be able to choose
11 from two or more types of context. As illustrated in the user interface 40 of the
12 application window of **Figure 2b**, by clicking on the context list box 12, the
13 user is presented with a context list 13 showing all of the contexts available in
14 the current window. In user interface 40, users can choose from: Clients,
15 Users, Devices and Media, Backup, Monitor, Restore, Reporting, and
16 Database. In this figure, the context list 13 is presented in an expanded
17 combination ("combo") box as shown. By selecting one of the contexts in the
18 list, the user moves to that context. If the user selected "Restore" from the
19 context list 13, he or she would be presented with the user interface in **Figure**
20 **2c**.

21
22 In **Figure 2c**, the context of the application window 50 has been
23 changed to "Restore", as shown in context list box 12 and scoping pane 14.

1 The available objects to restore are shown in content pane 16 and include:
2 Filesystem, OmniBack II database, Rawdisk, and SAP. **Figure 2d** illustrates
3 another context within the same window. "Devices and Media" is the current
4 context as shown in context list box 12. Selection of this context from the
5 context list (14 in **Figure 2b**) allows the user to see what backup devices are
6 available and what media have been configured for doing backups. The
7 devices and media are shown in scoping pane 14 of user interface 60. Notice
8 that different menu items 17 are available in the three contexts shown in
9 **Figures 2a, 2c, and 2d.**

10
11 In addition to being applicable to application windows that display
12 objects, the current invention is equally applicable to application windows that
13 present tasks. Referring now to Figures 3a and 3b, the use of the context list
14 box 12 in a task application window is illustrated. In **Figure 3a**, the context list
15 box 12 of the user interface 70 has been set to "Service Definition Tasks" and
16 the task "Modify Service" has been chosen in scoping pane 14. The scoping
17 pane in this context shows the hierarchy of tasks that are related to defining
18 the services that an IT group would provide its customers. The selected item
19 "Modify Service" is one of the many different service definition tasks. Content
20 pane 16 illustrates the properties box associated with modifying the service
21 named Order Processing System Access. **Figure 3b** presents the same task
22 application window "OV Tasks" with a different context. The context list box 12
23 has been set to OpenView Customization. The change in the context list box

1 results in setting the context for user interface 80 to OpenView Customization.
2 The scoping pane 14 now presents the hierarchy of the tasks associated with
3 customizing OpenView applications. Selecting "New Alarm Wizard" in the
4 scoping pane causes the content pane 16 to present a wizard containing the
5 fields for the data the user must provide to complete the task. Again, by
6 simply manipulating the context chosen in context list box 12, the user
7 explicitly filters information and is presented with that subset of the information
8 that is relevant to the context in the same application window.

9
10 For the figures shown above, a change in the context in which
11 information is presented to users will likely result in a change in the
12 functionality that is available in the menu bar 17 and toolbar buttons 18, the
13 objects, tasks, or tools available in the scoping pane 14 from which the user
14 may choose, and the views that can be presented in content pane 16 of the
15 user interface. The change in context can filter out functionality or object
16 capabilities associated with other perspectives or it can selectively make some
17 object capabilities or functionality less visible than the functionality associated
18 with the current perspective of an object but still available to the user. As can
19 be seen in user interface 90 of the application window of **Figure 4a**, the
20 context list box 12 has been set to "IP Internet." The pop-up menu 92 for
21 system "CltGeorg" has been accessed. In this context, all of the menu items
22 shown, which are one or more object capabilities corresponding to this
23 particular perspective of the selected object "CltGeorge," are related to the

1 network status. Additional capabilities, shown as "Additional Actions" menu
2 94, which are associated with another perspective of this object, are not
3 selected and thus not shown in this particular perspective view of the
4 "CltGeorg" object. In **Figure 4b**, the same application with a different context
5 and thus perspective is presented to the user. The context list box 12 has
6 been set to "IT Services". Note that user interface 100 shows the same
7 selected object "CltGeorg" from user interface 90 from Figure 4a but from the
8 new perspective of service management. The pop-up menu 102 for system
9 "CltGeorg" now shows only menu items associated with the services and
10 service status. None of the functionality associated with network management
11 is visible. Users can access functionality associated with other contexts by
12 clicking on the "Additional Actions.." menu item 104. This results in the
13 presentation of a separate pop-up dialog, user interface 110, that contains a
14 list box 112 with all of the actions (or capabilities) associated with object
15 "CltGeorg" from other contexts 114, as shown in **Figure 4c**. Note that the
16 menu items 92 from the IP Internet context in Figure 4a appear in this dialog
17 box as actions 116 under the Network management category. By separating
18 the functionality or object capability(ies) associated with the current contextual
19 perspective from functionality associated with other context perspectives,
20 users are not over burdened with irrelevant functionality. However, users are
21 conveniently able to access this functionality without the need to change
22 contexts.

1 It is envisioned that the information presented in content pane 16 may
2 take any form relevant to the data being presented, for example graphical,
3 tabular, properties, wizard or chart form. The content panes 16 of Figures 1a
4 and 1b, for instance, show information in graphical form; content pane 16 of
5 Figure 1a shows a network map, while Figure 1b shows a service topology.
6 The content panes 16 of Figures 2a, 2b, 2c, and 2d shown information in
7 tabular form. The content pane of Figure 3a shows a properties box while the
8 content pane of Figure 3b shows a wizard.

9
10 The present invention therefore provides many advantages to the user.
11 The user can focus on information that is relevant to the current task and this
12 information can in turn be shared by various applications as context filters in
13 order to vary the way that information is presented to the user. This enables
14 the information presentation of the user interface to match the preferences and
15 needs of the user as the user interacts with the computer system. Moreover,
16 the sharing of contextual information allows independently developed
17 applications to appear to the user as highly integrated applications that work
18 closely together within a single user interface window in the computer
19 environment.

20
21 The judicious manipulation of context can also enhance the productivity
22 of the user. Contextual information such as the currently visible view or recent
23 user actions can be used to set defaults that minimize the need for user input.

1 The contextual information can be used to filter irrelevant information or steps
2 so that users can more quickly reach their goals. The passing of contextual
3 information between software within an application and between applications
4 can eliminate the need for the user to duplicate their previous actions.

5
6 While the invention has been particularly shown and described with
7 reference to a preferred embodiment, it will be understood by those skilled in
8 the art that various changes in form and detail may be made therein without
9 departing from the spirit and scope of the invention.

100200122-118